

**【特許請求の範囲】**

【請求項 1】 ネットワーク上のパケットを監視し、予め用意されたシナリオに基づいて、その通信プロトコルを試験する通信プロトコル試験装置において、前記シナリオには、試験条件および前記試験条件が満足されたときに実行される動作を定義されたシナリオ行が登録され、前記試験条件が満足されると前記シナリオに登録された動作が実行され、前記試験条件が満足されないと通信プロトコルで規定された標準動作が実行されることを特徴とする通信プロトコル試験装置。

【請求項 2】 前記シナリオはヘッダ部およびコンテンツ部を含み、前記ヘッダ部には、試験対象の通信コネクションを識別する第 1 条件が登録され、前記コンテンツ部には、複数のシナリオ行が時系列に登録され、前記各シナリオ行には、パケットを選別する第 2 条件および前記第 2 条件が満足されたときに実行される動作が定義され、前記第 1 条件を満足する通信コネクションにおいて前記第 2 条件が満足されると、前記各シナリオ行に登録された動作が実行され、それ以外では通信プロトコルで規定された標準動作が実行されることを特徴とする請求項 1 に記載の通信プロトコル試験装置。

【請求項 3】 前記シナリオ行において、前記第 2 条件が満足されたときに実行される動作として、内部変数を更新するか否かが定義されていることを特徴とする請求項 2 に記載の通信プロトコル試験装置。

【請求項 4】 前記第 1 および第 2 条件が満足されたときに歩進するシナリオ行カウンタを具備し、前記カウンタのカウント値に基づいて各シナリオ行が順番に実行されることを特徴とする請求項 2 または 3 に記載の通信プロトコル試験装置。

【請求項 5】 通信ログを記憶する手段をさらに具備し、前記通信ログは、送受されたパケットの情報と共に、前記シナリオ行カウンタのカウント値および送受信を中止したパケットの記録の少なくとも一方を記憶することを特徴とする請求項 4 に記載の通信プロトコル試験装置。

【請求項 6】 前記通信プロトコルが TCP であることを特徴とする請求項 1 ないし 5 のいずれかに記載の通信プロトコル試験装置。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】 本発明は、通信プロトコル試験装置に係り、特に、シナリオ記述に基づいて各種の通信プロトコルを試験する通信プロトコル試験装置に関する。

**【0002】**

【従来の技術】 ネットワークを介して接続された端末同

士は、所定の通信プロトコルにしたがって動作することで情報を相互に送受し合う。このような装置の通信プロトコルを試験するために、従来から通信プロトコルの試験装置が用いられている。通信プロトコル試験装置は、被試験システムに送信したメッセージに対する応答の有無やそのメッセージの内容等をチェックすることで、被試験システムに実装されている通信プロトコルが正常か否かを調べる。従来から種々の通信プロトコル試験装置が提案されており、たとえば以下のようなものがあった。

**(1) 通信プロトコル適合性試験装置**

【0003】 特開平 7-245641 号公報では、OSI 通信システムを対象に、送受信の PDU (Protocol Data Unit) を全てシナリオに記述して試験を実行する通信プロトコル適合性試験装置が提案されている。送信 PDU を処理する場合は、設定されたパラメータの情報も含め、その PDU の送信を行う。受信 PDU を処理する場合は、その PDU が受信されたかどうかを確認し、PDU のパラメータも含めて一致するか否かの判定を行う。

**(2) トラフィックジェネレータ**

【0004】 IP アドレスや TCP のポート番号など一部のパラメータが固定され、他のパラメータがランダムに設定されるパケットを大量に生成・送出してネットワークのパフォーマンスを測定するトラフィックジェネレータとして、例えば IXIA (<http://www.ixiacom.com>) が知られている。IXIA では、パケットの送信が一定周期あるいはランダムな周期で行われる。

**(3) tcp などの TCP 試験ツール**

【0005】 tcp (RFC2398 の 2.11. 参照) は、TCP 上で動作するアプリケーションであり、パケットのサイズや送受信バッファサイズなどを設定して、TCP のデータ転送に関するパフォーマンスを測定する。

**【0006】**

【発明が解決しようとする課題】 通信プロトコル適合性試験装置におけるシナリオ記述方法は、一連の通信シーケンスを全て記述しなければならない。したがって、通信シーケンスが長く、大量の PDU を送受信しなければならない場合には多大な労力が要求される。また、通信プロトコル適合性試験装置におけるシナリオ記述方法では、TCP のように、トランスポート層の試験を行う場合、アプリケーションのデータを全て設定しなければ、相手側の通信システムの TCP を試験することができない。

【0007】 IXIA などのトラフィックジェネレータを用いることにより、任意の TCP のパケットを作成することは可能である。しかしながら、パケットの受信処理に関しては、パケット数をカウントするのみである。従って、パケットの受信と、パケットの送信とを同期させることができない。このため、トラフィックジェネレータは、TCP のプロトコル動作の試験を行うのには適していない。

10

20

30

40

50

【0008】また、トラフィックジェネレータでは、例えば「特定のシーケンス番号の packets を送信しない」、「あるシーケンス番号を受信したら 10 秒間待機する」などといった高度な設定はできない。

【0009】tcp は、正常なプロトコル動作において TCP のパフォーマンスを測定できるものの、故意に再送や packets 損失を発生させたり、プロトコル動作に合わない packets を送信したりすることができない。

【0010】本発明の目的は、上記した従来技術の課題を解決し、試験に関するシナリオのみを記述するだけで通信プロトコルの実装の問題点を試験できる通信プロトコル試験装置を提供することにある。

【0011】

【課題を解決するための手段】上記した目的を達成するために、本発明は、ネットワーク上の packets を監視し、予め用意されたシナリオに基づいて、その通信プロトコルを試験する通信プロトコル試験装置において、シナリオには、試験条件および前記試験条件が満足されたときに実行される動作が定義されたシナリオ行を登録し、試験条件が満足されると前記シナリオに登録された動作を実行し、試験条件が満足されないと通信プロトコルで規定された標準動作を実行するようにした点に特徴がある。

【0012】上記した特徴によれば、シナリオには所望の試験条件や試験内容のみを記述すれば良いので、シナリオの作成に要する労力が低減される。

【0013】

【発明の実施の形態】以下、図面を参照して本発明の好ましい実施の形態について説明する。図 1、2 は、本発明を適用した TCP 試験装置のネットワークへの適用例を示したブロック図である。

【0014】本発明の TCP 試験装置 1 は、被試験対象のシステム 2 と ftp や tcp など汎用的な TCP 通信アプリケーションを用いて試験用の TCP 通信を行う。被試験システム 2 は、図 1 に示したように、LAN 上に存在する場合（形態 1）と、図 2 のように、インターネット 3 を介した遠隔に存在する場合（形態 2）とが考えられる。図 1 の形態 1 では、TCP 試験装置 1 および被試験システム 2 の一方がクライアント、他方がサーバとして振る舞い、双方のシステムから TCP の通信を開始することができる。

【0015】図 2 の形態 2 では、TCP 試験装置 1 が被試験システム 2 を制御できないことを前提とし、TCP 試験装置 1 はクライアントとしてのみ動作し、被試験システム 2 上で動作しているサーバプログラム（例えば、ftp サーバなど）と、TCP の通信を行うことが可能である。いずれの場合においても、被試験対象システム 2 の TCP の実装が試験対象となる。

【0016】図 3 は、上記した TCP 試験装置 1 の構成を示したブロック図であり、LAN などのネットワークと TCP

試験装置 1 との通信を制御するネットワークインターフェース (I/F) 11 と、TCP 処理機能およびシナリオに基づく試験機能を備えた TCP 処理部 10 と、TCP 試験装置 1 の主要動作を制御する制御部 12 と、マン/マシンインターフェースとしての表示部 13 および操作部 14 とを含む。

【0017】前記 TCP 処理部 10 において、シナリオ記憶部 101 には、後に図 4 を参照して詳述するように、試験条件および前記試験条件が満足されたときに実行される動作を定義した複数のシナリオ行が時系列に登録されている。シナリオ処理部 102 は、このシナリオ記憶部 101 に記憶されたシナリオに基づいて通信プロトコルを試験する。TCP 実行部 103 は、TCP で規定された標準動作を実行する。通信ログ記憶部 104 は、試験期間中に送受された packets のシーケンスを記憶する。

【0018】前記シナリオ処理部 102 において、通信コネクション識別部 1021 は、TCP コネクションが試験対象のコネクションであるか否かを識別する。パケット選別部 1022 は、試験対象のコネクションにおいて送受される packets が試験対象の packets であるか否かを判別する。シナリオ実行部 1023 は、試験対象の TCP コネクションにおける試験対象の packets に対して、シナリオ行カウンタ 1023a のカウント値で指示されるシナリオ行に登録された処理を実行する。それ以外の packets に対しては、前記 TCP 実行部 103 により通常の TCP が実行される。

【0019】制御部 12 において、試験環境設定部 121 は、所望の試験内容に応じたシナリオのシナリオ記憶部 101 へのアップロード、既にアップロードされているシナリオの参照、修正および削除、ならびにログの蓄積要求を実行する。ログ解析部 123 は、前記通信ログ記憶部 104 に記憶されたログを解析して表示部 13 へ表示する。

【0020】試験実行部 122 は、ftp や tcp など汎用的な TCP アプリケーションを実行し、TCP 処理部 10 に対してアプリケーションに基づく packets 送信を要求する。ネットワーク構成が図 1 の形態 1 の場合は、双方のシステムで試験用アプリケーションを実行する。ネットワーク構成が図 2 の形態 2 の場合は、TCP 試験装置 1 において、遠隔の被試験システムで動作しているサーバアプリケーションと通信可能な試験用のアプリケーションを起動する。

【0021】次いで、図 4 を参照してシナリオの記述方法について説明する。本実施形態では、シナリオがヘッダ部とコンテンツ部とで構成され、汎用のテキストエディタ等を用いて記述することができる。

【0022】ヘッダ部 <header> には、試験対象とする TCP コネクションを識別するための情報（第 1 条件）と、その TCP コネクションの中で適用されるコネクション確立時のアクションとが設定される。

【 0 0 2 3 】 TCPコネクション情報としては、発IPアドレス (src-addr)、着IPアドレス (dst-addr)、発ポート番号 (src-port)、着ポート番号 (dst-port) を指定可能とし、この条件に該当するTCPコネクションのみが、シナリオに従ったTCP通信を行う。これらの項目の指定はオプションとし、指定された項目のみが条件判断されることになる。

【 0 0 2 4 】 TCPコネクションの中で適用されるコネクション確立時のアクションとしては、コネクション確立要求パケット (SYN) またはコネクション確立確認パケット (SYN+ACK) のTCPオプションとして使用される項目を設定する。図 4 では、MSS (Maximum Segment Size) 値を 1 0 としたMSSオプション (mss(10))、SACK permittedオプション (sack)、タイムスタンプオプション (timestamp)、WSF (WindowScale Factor) 値を 0 としたウィンドウスケールオプション (wsf(0)) が設定されている。

【 0 0 2 5 】 コンテンツ部<content>には、TCPコネクション確立後のTCPの振舞いが記述される。コンテンツ部の各ステップ (以下、シナリオ行と表現する) には、シナリオを実行するための条件 (第 2 条件) と、この条件に合致した場合に実行されるアクションとが記述されている。条件とアクションとの区切りにはセミコロン (;) が挿入される。以下、各シナリオ行の解釈方法について説明する。

(1) シナリオ行の第 1 行

「recv seq=1 var-upd=ON ; send seq=1 ack=11 flag=(ack) win=20 var-upd=ON」

【 0 0 2 6 】 シーケンス番号が 1 (seq=1) のパケットを受信 (recv) した場合に、受信処理に伴う内部変数の更新を行い (var-upd=ON)、シーケンス番号が 1 (seq=1)、応答確認番号が 1 1 (ack=11)、フラグビットがACKフラグのみ (flag=(ack))、ウィンドウサイズが 2 0 (win=20) のパケットを送信し、送信処理に伴う内部変数

(5) 同第 7 - 1 0 行

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

「recv seq=31 var-upd=ON ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON」

【 0 0 3 0 】 シーケンス番号が 3 1 (seq=31) のパケットを受信 (recv) した場合に、受信処理に伴う内部変数の更新を行い (var-upd=ON)、シーケンス番号が 1 (seq=1)、応答確認番号が 2 1 (ack=21)、SACK (Selective Acknowledgment) パラメータが 3 1 - 4 1 (sack=(31-41))、フラグビットがACKフラグのみ (flag=(ack))、ウ

の更新を行う (var-upd=ON)。

(2) 同第 2 行

「recv seq=11 var-upd=OFF ; ignore」

【 0 0 2 7 】 シーケンス番号が 1 1 (seq=11) のパケットを受信 (recv) した場合に、受信処理に伴う内部変数の更新を行わず (var-upd=OFF)、送信処理も行わない (ignore)。この場合、被試験システムが送信した本パケットがネットワーク上で紛失し、TCP試験装置に届かなかった場合と同じ状態となる。従って、このシナリオ行により、ネットワーク上でのパケット紛失を再現できる。

(3) 同第 3 行

「send seq=1 ack=11 var-upd=yes ; ignore」

【 0 0 2 8 】 シーケンス番号が 1 (seq=1) であり、かつ応答確認番号が 1 1 (ack=11) のパケットを送信 (send) した場合に、送信処理に伴う内部変数の更新は行わ (var-upd=ON)、送信処理は行わない (ignore)。この場合、TCP試験装置 1 が送信した本パケットがネットワーク上で紛失し、被試験システム 2 に届かなかった場合と同じ状態となる。従って、このシナリオ行により、ネットワーク上でのパケット紛失を再現できる。

(4) 同第 4 行

「wait 1 ; send seq=1 ack=11 flag=(ack) win=30 var-upd=OFF」

【 0 0 2 9 】 1 タイムスロット (500ミリ秒) 待機した後、シーケンス番号が 1 (seq=1)、応答確認番号が 1 1 (ack=11)、フラグビットがACKフラグのみ (flag=(ack))、ウィンドウサイズが 3 0 (win=30) のパケットを送信する。ただし、送信処理に伴う内部変数の更新は行わない (var-upd=OFF)。シナリオ行の第 4、5 行との連携により、被試験システムに対してACKパケットを 1 タイムスロット分だけ遅らせて届くように制御することができる。

ウィンドウサイズが 3 0 (win=30) のパケットを 4 つ送信し、送信処理に伴う内部変数の更新を行う (var-upd=ON)。

(6) 同第 1 1 行

「recv seq>70 var-upd=ON ; send seq=1 flag=(rst) win=30 var-upd=ON」

【 0 0 3 1 】 シーケンス番号が 7 0 よりも大きい (seq>

70) パケットを受信 (recv) した場合、受信処理に伴う内部変数の更新を行い (var-upd=ON)、シーケンス番号が 1 (seq=1)、フラグビットが RST フラグのみ (flag=(rst))、ウィンドウサイズが 30 (win=30) のパケットを送信し、送信処理に伴う内部変数の更新を行う (var-upd=ON)。条件には、イコールの他に、4 種類の不等号 (>, <, >=, <=) を用いることができる。本パケットの送信により、被試験システムに、TCP コネクションの解放が通知される。

【0032】図 4 のシナリオを実行した場合の通信シーケンスを図 5 に示す。被試験システム 2 では、DATA 11-20 のパケット紛失により、タイムアウト再送が発生し、重複した ACK 21 の受信により、fast retransmit による再送が発生している。fast retransmit による DATA 21-30 から DATA 71-80 までのシーケンスは、通常の TCP の処理に基づいている。

【0033】次いで、本実施形態の動作を、図 6, 7, 8 のフローチャートを参照して説明する。各フローチャートは、主に TCP 処理部 10 の動作を示しており、TCP 処理部 10 は、パケットの送受信が検出されるか、あるいは後述する WAIT タイマがタイムアウトしたことに応答して起動される。

【0034】前記制御部 12 の試験実行部 122 が所定の TCP アプリケーションを起動してパケットの送受が開始され、これが TCP 処理部 10 においてイベントとして検知されると、図 6 の処理がスタートする。

【0035】ステップ S1 では、検出されたパケットが試験対象の TCP コネクションにおいて送受されたものであるか否かが、前記通信コネクション識別部 1021 において判定される。試験対象の TCP コネクションであると、ステップ S2 では、TCP コネクションが確立済みであるか否かが判定される。確立済みであれば、ステップ S3 において、現在のシナリオ行の条件が WAIT であるか否かが判定される。

【0036】現在のシナリオ行は、前記シナリオ行カウンタ 1023a によりカウントされている。条件が WAIT であれば、ステップ S13 において、500 ミリ秒でタイムアウトする WAIT タイマがスタートする。条件が WAIT でなければ更に、ステップ S4 において、シナリオが終了しているか否かが判定される。

【0037】シナリオ行が終了していなければ、ステップ S5 において、受信パケットあるいは送信パケットが現在のシナリオ行の条件に一致するか否かが判定される。すなわち、今回のシナリオが前記図 4 の通りであり、現在のシナリオ行が第 1 行であれば、パケットのシーケンス番号が 1 (seq=1) であるか否かが判定される。なお、複数の条件が設定されている場合には、全ての条件の論理積に基づいて判定される。条件が不一致であれば、ステップ S14 において、シナリオ行カウンタ 1023a を変更することなく通常の TCP 処理が実行され

る。

【0038】これに対して、条件が一致していると判定されれば、ステップ S6 では、条件部分に内部変数の更新が指定されている場合に、条件部分のパケットの送受信に伴って内部変数が更新される。ステップ S7 では、アクションが ignore (無視) であるか否かが判定される。アクションが ignore であれば、ステップ S12 においてシナリオ行カウンタ 1023a をインクリメント (歩進) させた後、今回の処理を終了して次のイベント発生を待つ。

【0039】これに対して、アクションが ignore でなければ、ステップ S8 において、アクションに指定されたパケットが送信される。前記第 1 行であれば、シーケンス番号が 1 (seq=1)、応答確認番号が 11 (ack=11)、フラグビットが ACK フラグのみ (flag=(ack))、ウィンドウサイズが 20 (win=20) のパケットが送信される。ステップ S9 では、アクション部分に内部変数の更新が指定されている場合に、アクションに伴って内部変数が更新される。

【0040】ステップ S10 ではシナリオ行のカウンタ 1023a がインクリメントされる。ステップ S11 では、シナリオ行に条件が記述されているか否かに基づいて、継続するアクションの有無が判別される。図 4 の第 8, 9, 10 行のように、シナリオ行に条件が記述されていなければ、継続するアクションがあると判定されてステップ S8 へ戻る。シナリオ行に条件が記述されていれば、今回の処理を終了して次のイベント発生を待つ。

【0041】一方、前記ステップ S2 において、コネクションが未だに確立されていないと判定されると、図 7 のステップ S21 において、SYN (コネクションの確立要求) または SYN+ACK (コネクションの確立要求確認) の送信であるか否かが判定される。SYN または SYN+ACK の送信であれば、ステップ S22 において、ヘッダ部の syn-opt で指定された TCP オプションが設定される。ステップ S23 では、SYN または SYN+ACK が送信される。

【0042】前記ステップ S21 において、SYN または SYN+ACK の送信ではないと判定され、さらにステップ S24 において、ACK 送信または ACK 受信と判定されるとステップ S25 へ進む。ステップ S25 では、コネクションが確立済みとされ、ステップ S26 では、シナリオ行カウンタ 1023a が「1」に設定される。ステップ S27 では、通常の TCP に基づく処理が実行される。

【0043】次いで、前記ステップ S13 でスタートした WAIT タイマがタイムアウトしたことによるイベントが発生した場合の動作を、図 8 のフローチャートを参照して説明する。

【0044】ステップ S31 では、アクションで指定されたパケットの送信が行われる。ステップ S32 では、アクション部分に内部変数の更新が指定されている場合に限り、アクションに伴う内部変数の更新が行われる。

【0045】ステップS33では、シナリオ行のカウンタ1023aがインクリメントされる。ステップS34では、シナリオ行に条件が記述されているか否かに基づいて、継続するアクションの有無が判別される。シナリオ行に条件が記述されていないければ、継続するアクションがあると判定されてステップS31へ戻る。シナリオ行に条件が記述されていれば、今回の処理を終了して次のイベント発生を待つ。

【0046】図9は、図4のシナリオに基づいて実行された試験期間中に収集された通信ログの一例を示している。通信ログでは、1行につき一つの送受信パケットに関する情報が記述されている。各行に記録される情報は、シナリオ行カウンタ1023aのカウンタ値、パケット発生時刻（最初のパケットからの相対時刻：単位はミリ秒）、フラグビット種別、シーケンス番号等の各パラメータ、内部変数の更新の行った否かの識別子（行った場合は、var-updが付記される）から構成される。このように、シナリオ行カウンタ1023aのカウンタ値を併記することにより、どのシナリオ行を実行中であったかを簡単に認識できるようになる。

【0047】さらに、本実施形態では受信パケットに対してインデントを行うことにより、送受信パケットのいずれであるかが区別される。また、実際に送受信されなかったパケットは括弧内に表示されるので、送受信されたパケットと送受信されなかったパケットとの識別が容易になる。

【0048】なお、上記した実施形態では、本発明をTCP試験装置を例にして説明したが、本発明はこれのみに限定されるものではなく、他の通信プロトコルにも同様に適用することができる。

【0049】

【発明の効果】本発明によれば、以下のような効果が達成される。

(1)試験手順はシナリオ記述により指定し、誤った振舞いを行いたい箇所のみを、シナリオに記述すれば良いので、従来の試験システムのように、通信シーケンス全体をシナリオに記述しなければならない場合に比べて労力が軽減される。

(2)試験装置は、アプリケーションとは独立に設計され

ているため、ftpやttcpなどの既存のアプリケーションを用いてTCPの試験を行うことができる。

(3)特定のパケットを受信しない、特定のパケットを送受信しないというシナリオを記述することにより、ネットワーク上でのパケット紛失をシミュレーションすることができる。

(4)特定のIPアドレスまたは特定のポート番号の通信のみを試験対象として設定することにより、telnetやXウィンドウなどの通信アプリケーションにおいて通常のTCP通信を行いながら、特定のTCPコネクションに対してのみ試験を選択的に行うことができる。

(5)通信ログにシナリオ行カウンタのカウンタ値が表示されるので、どのシナリオ行を実行することにより得られたログであるかを簡単に認識できる。

(6)通信ログでは、実際に送受信されなかったパケットが、送受信されたパケットと区別して表示されるので、両者の識別が容易になる。

【図面の簡単な説明】

【図1】 本発明を適用したTCP試験装置のネットワークへの適用例（第1形態）を示したブロック図である。

【図2】 本発明を適用したTCP試験装置のネットワークへの適用例（第2形態）を示したブロック図である。

【図3】 TCP試験装置の第1実施形態のブロック図である。

【図4】 シナリオの一例を示した図である。

【図5】 図4のシナリオを実行した場合の通信シーケンスを示した図である。

【図6】 本発明の動作を示したフローチャート（その1）である。

【図7】 本発明の動作を示したフローチャート（その2）である。

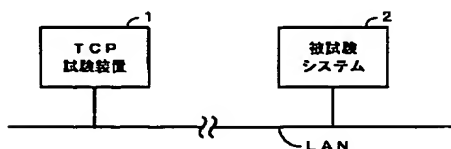
【図8】 本発明の動作を示したフローチャート（その3）である。

【図9】 通信ログの一例を示した図である。

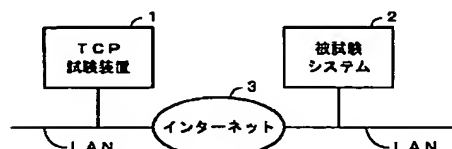
【符号の説明】

1…TCP試験装置、2…被試験システム、3…インターネット、10…TCP処理部、11…ネットワークインターフェース、12…制御部、13…表示部、14…通信ログ記憶部

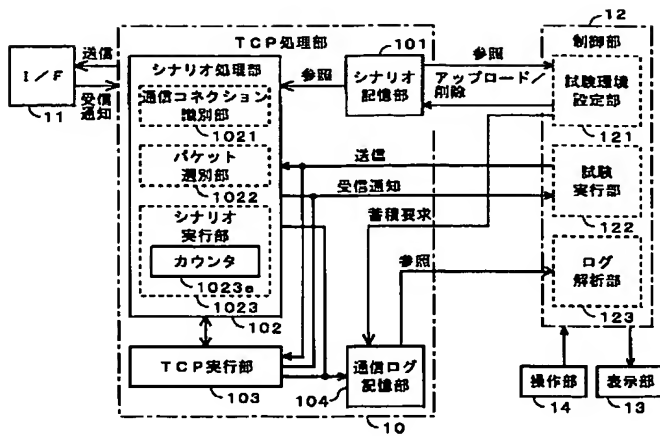
【図1】



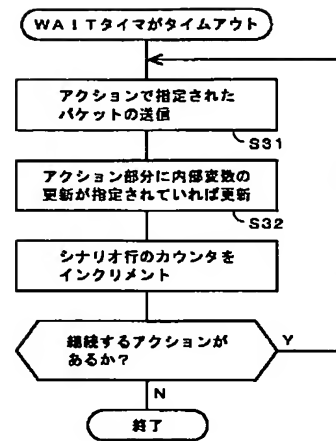
【図2】



【図 3】



【図 8】



【図 4】

```
<header>
src-addr=192.168.0.1
dst-addr=192.168.0.2
dst-port=20
syn-opt=mss(10), sack, timestamp, wsf(0)
```

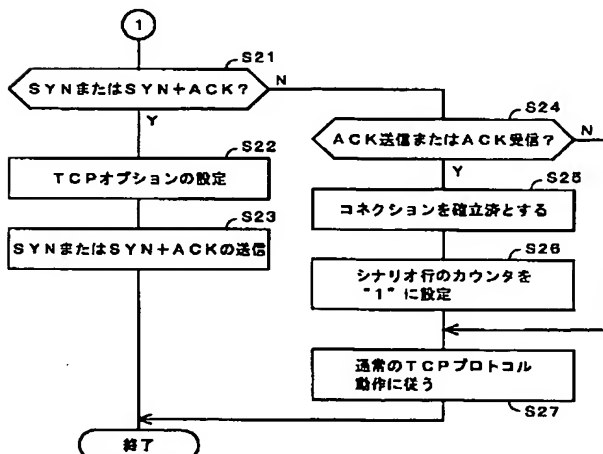
&lt;content&gt;

```

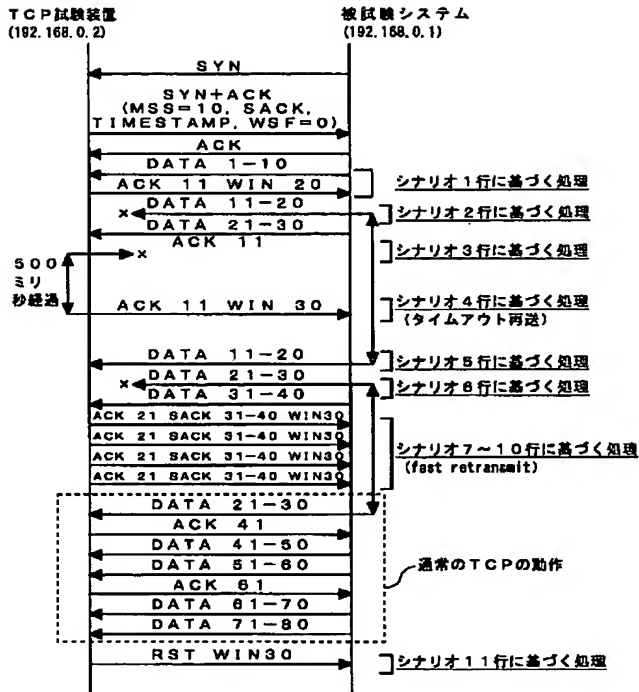
第 1 行..... rcv seq=1 var-upd=ON      ; send seq=1 ack=11 flag=(ack) win=20 var-upd=ON
第 2 行..... rcv seq=11 var-upd=OFF     ; ignore
第 3 行..... send seq=1 ack=11 var-upd=yes ; ignore
第 4 行..... wait 1                      ; send seq=1 ack=11 flag=(ack) win=30 var-upd=OFF
第 5 行..... rcv seq=11 var-upd=ON      ; ignore
第 6 行..... rcv seq=21 var-upd=OFF     ; ignore
第 7 行..... rcv seq=31 var-upd=ON      ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON
第 8 行.....                               ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON
第 9 行.....                               ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON
第 10 行.....                               ; send seq=1 ack=21 sack=(31-41) flag=(ack) win=30 var-upd=ON
第 11 行..... rcv seq>70 var-upd=ON     ; send seq=1 flag=(rst) win=30 var-upd=ON

```

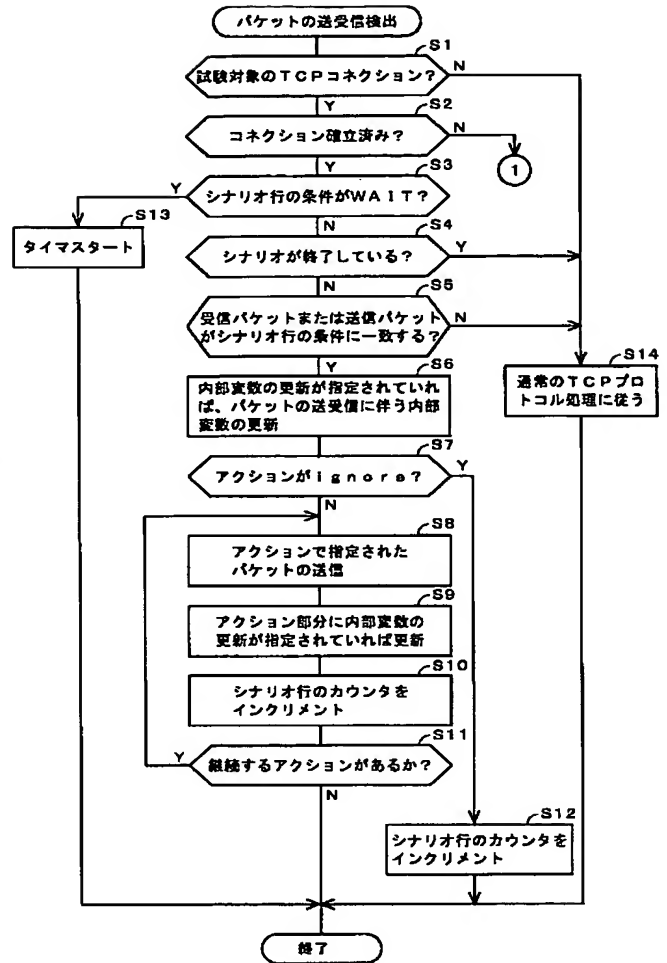
【図 7】



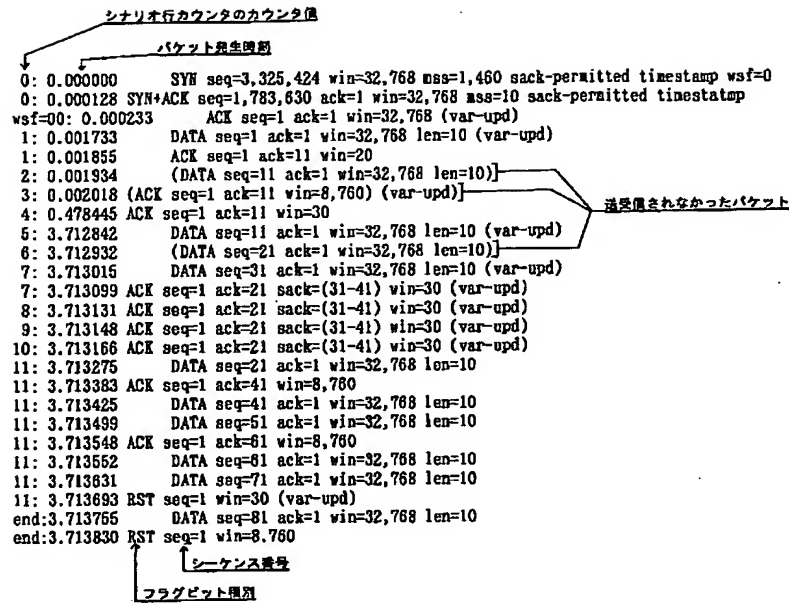
【図 5】



【図 6】



【図 9】



フロントページの続き

(72)発明者 加藤 聡彦

埼玉県上福岡市大原二丁目 1 番15号 株式  
会社ケイディーディーアイ研究所内

F ターム(参考)

5B089 GB02 KA12 KB04 MC15

5K034 AA19 EE11 FF01 FF02 HH63

JJ24 LL01 TT02

5K035 AA07 CC01 CC03 EE13 FF01

FF02 HH02 HH07